A numerical method for calculating the Green's function arising from electronic structure theory

T. Sogabe[1] T. Hoshi[2,3] S.-L. Zhang[1], T. Fujiwara[2,3]

[1] *Department of Computational Science and Engineering, Nagoya University, Furo-cho, Chikusa-ku,Nagoya 464-8603, Japan*

[2] *Department of Applied Physics, The University of Tokyo, Hongo, 7-3-1, Bunkyo-ku, Tokyo, 113-8656, Japan*

[3] *Core Research for Evolutional Science and Technology, Japan Science and Technology Agency (CREST-JST), 4-1-8 Honcho, Kawaguchi-shi, Saitama 332-0012, Japan*

February 2006

Abstract

We developed a fast numerical method for complex symmetric shifted linear systems, which is motivated by the quantum-mechanical (electronic-structure) theory in nanoscale materials. The method is named shifted Conjugate Orthogonal Conjugate Gradient (shifted COCG) method. The formulation is given and several numerical aspects are discussed.

1 Introduction

The quantum-mechanical feature of electrons plays a crucial role in nanoscale materials and its mathematical foundation is reduced to linear-algebraic problems with given large matrices H, called Hamiltonian. The physical properties of electrons can be described by the Green's function G that is defined as inversed matrix $G(z) := (zI - H)^{-1}$ with a complex variable z whose real part corresponds to energy. Since the standard matrix-inversion procedure requires an impractical computational cost in case of large matrices or in nanoscale materials, there is a strong need for the fast solution of the Green's function, see the references in [3].

Here we introduce a new method for calculating the Green's function [3]. Let H be an N-by-N real symmetric Hamiltonian matrix, then any element of the Green's function can be written as

$$G_{ij}(z) = \boldsymbol{e}_i^T (zI - H)^{-1} \boldsymbol{e}_j, \tag{1}$$

where $\boldsymbol{e}_i$ denotes the ith unit vector, and the complex energy $z = \sigma + i\delta \in C$. Note that the i, j entry of (1) can be obtained by two steps: first, compute

1

$(zI - H)\boldsymbol{x} = \boldsymbol{e}_j$, and then compute $\boldsymbol{e}_i^T \boldsymbol{x}$. Since an integral of $G_{ij}(z)$ with respect to σ is needed to obtain any physical quantity [3], the numerical integration requires $\{(\sigma_k + i\delta)I - H\}\boldsymbol{x}^{(k)} = \boldsymbol{e}_j$ for $k = 1, \dots, M$. Hence, the problem reduces to solving the following shifted linear systems with complex symmetric matrices:

$$(A + \sigma_k I)\boldsymbol{x}^{(k)} = \boldsymbol{b} \quad \text{for} \quad k = 1, \dots, M. \tag{2}$$

The paper is organized as follows: in the next section, we describe the algorithm and the property of COCG for solving complex symmetric linear systems. In §3, to solve (2) efficiently, a numerical method named shifted COCG is proposed and seed switching technique is introduced. In §4, we report some numerical experiments. Finally, we make some concluding remarks in §5.

2 The COCG method

Matrix A is called complex symmetric if A is not Hermitian but symmetric $A = A^T \neq A^H$. To solve the linear systems, the COCG method [4] has been proposed and is known as one of the most successful Krylov subspace methods.

Algorithm 1: COCG

$\boldsymbol{x}_0$ is an initial guess,

$\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0, \ \boldsymbol{p}_{-1} = \boldsymbol{0}, \ \beta_{-1} = 0,$

for $n = 0, 1, \dots$ until $\|\boldsymbol{r}_n\| \leq \epsilon_1 \|\boldsymbol{b}\|$ **do:**

    $\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_{n-1}\boldsymbol{p}_{n-1},$

    $\alpha_n = \dfrac{\boldsymbol{r}_n^T \boldsymbol{r}_n}{\boldsymbol{p}_n^T A \boldsymbol{p}_n},$

    $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n \boldsymbol{p}_n,$

    $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n - \alpha_n A \boldsymbol{p}_n,$

    $\beta_n = \dfrac{\boldsymbol{r}_{n+1}^T \boldsymbol{r}_{n+1}}{\boldsymbol{r}_n^T \boldsymbol{r}_n}.$

end

Observing Algorithm 1, we see that the nth residual can be written as $\boldsymbol{r}_n(:= \boldsymbol{b} - A\boldsymbol{x}_n) = R_n(A)\boldsymbol{r}_0$, where $R_0(\lambda) = 1$, $R_1(\lambda) = (1 - \alpha_0 \lambda)R_0(\lambda)$, and

$$R_n(\lambda) = \left(1 + \frac{\beta_{n-2}}{\alpha_{n-2}}\alpha_{n-1} - \alpha_{n-1}\lambda\right)R_{n-1}(\lambda) - \frac{\beta_{n-2}}{\alpha_{n-2}}\alpha_{n-1}R_{n-2}(\lambda). \tag{3}$$

It is known that if breakdown does not occur, then the nth residual satisfies

$$\boldsymbol{r}_n \perp \overline{K_n(A, \boldsymbol{r}_0)}, \tag{4}$$

which leads to conjugate orthogonality $\boldsymbol{r}_i \perp \overline{\boldsymbol{r}}_j$ for $i \neq j$.

2

3   A shifted COCG method

In this section, we develop the COCG method for solving complex symmetric shifted linear systems. First, let us describe the the following theorem:

Theorem 1 (Frommer [2, Theorem 1]) *Let $W_1 \subseteq W_2 \subseteq \ldots \subseteq W_k$ be a sequence of nested subspaces of C^N such that W_n has dimension n and $W_n \cap (K_{n+1}(A, \boldsymbol{b}))^\perp = \{0\}$, $n = 1, \ldots, k$. Let $\boldsymbol{r}_n := R_n(A)\boldsymbol{b}$, $\boldsymbol{r}_n^\sigma := R_n^\sigma(A + \sigma I)\boldsymbol{b}$ be residual vectors satisfying*

$$\boldsymbol{r}_n, \boldsymbol{r}_n^\sigma \perp W_n, \quad n = 1, \ldots, k. \tag{5}$$

Then $\boldsymbol{r}_n$ and $\boldsymbol{r}_n^\sigma$ are collinear.

Corollary 1 *Let $\boldsymbol{r}_n$ and $\boldsymbol{r}_n^\sigma$ be the residual vectors of COCG started with $\boldsymbol{x}_0 = \boldsymbol{x}_0^\sigma = 0$. Then, $\boldsymbol{r}_n$ and $\boldsymbol{r}_n^\sigma$ are collinear, i.e., there exists $\pi_n^\sigma \in C$ such that $\boldsymbol{r}_n = \pi_n^\sigma \boldsymbol{r}_n^\sigma$.*

Proof. Since it follows from (4) that the COCG residuals satisfy (5) with $W_n = K_n(A, \boldsymbol{b}) = K_n(A + \sigma I, \boldsymbol{b})$, this result follows from Theorem 1.

Next, we give the formulas for computing $\boldsymbol{r}_{n+1}^\sigma$ by using the information of $\boldsymbol{r}_{n+1}$. It follows from the polynomial (3) that we have

$$\boldsymbol{r}_{n+1} = \left(1 + \frac{\beta_{n-1}}{\alpha_{n-1}}\alpha_n - \alpha_n A\right)\boldsymbol{r}_n - \frac{\beta_{n-1}}{\alpha_{n-1}}\alpha_n \boldsymbol{r}_{n-1}, \tag{6}$$

$$\boldsymbol{r}_{n+1}^\sigma = \left(1 + \frac{\beta_{n-1}^\sigma}{\alpha_{n-1}^\sigma}\alpha_n^\sigma - \alpha_n^\sigma(A + \sigma I)\right)\boldsymbol{r}_n^\sigma - \frac{\beta_{n-1}^\sigma}{\alpha_{n-1}^\sigma}\alpha_n^\sigma \boldsymbol{r}_{n-1}^\sigma. \tag{7}$$

Substituting the relation $\boldsymbol{r}_n = \pi_n^\sigma \boldsymbol{r}_n^\sigma$ into the previous recurrence (7), we have

$$\boldsymbol{r}_{n+1} = \left(1 + \frac{\beta_{n-1}^\sigma}{\alpha_{n-1}^\sigma}\alpha_n^\sigma - \alpha_n^\sigma(A + \sigma I)\right)\frac{\pi_{n+1}^\sigma}{\pi_n^\sigma}\boldsymbol{r}_n - \frac{\beta_{n-1}^\sigma \alpha_n^\sigma \pi_{n+1}^\sigma}{\alpha_{n-1}^\sigma \pi_{n-1}^\sigma}\boldsymbol{r}_{n-1}. \tag{8}$$

To obtain the computational formula for $\boldsymbol{r}_{n+1}^\sigma$, the three parameters α_n^σ, β_{n-1}^σ, and π_{n+1}^σ are essentially required. Hence, we give the formulas for the three parameters. First, comparing the coefficients of $A\boldsymbol{r}_n$ in (6) and (8), we find

$$\alpha_n^\sigma = \left(\pi_n^\sigma/\pi_{n+1}^\sigma\right)\alpha_n. \tag{9}$$

Second, comparing the coefficients of $\boldsymbol{r}_{n-1}$ leads to $\frac{\beta_{n-1}}{\alpha_{n-1}}\alpha_n = \frac{\beta_{n-1}^\sigma \alpha_n^\sigma \pi_{n+1}^\sigma}{\alpha_{n-1}^\sigma \pi_{n-1}^\sigma}$. Substituting the result of (9) into the previous equation, we have

$$\beta_{n-1}^\sigma = \left(\pi_{n-1}^\sigma/\pi_n^\sigma\right)^2 \beta_{n-1}. \tag{10}$$

Finally, comparing the coefficients of $\boldsymbol{r}_n$, we find $(1 + \frac{\beta_{n-1}^\sigma}{\alpha_{n-1}^\sigma}\alpha_n^\sigma - \alpha_n^\sigma \sigma)\frac{\pi_{n+1}^\sigma}{\pi_n^\sigma} = 1 + \frac{\beta_{n-1}}{\alpha_{n-1}}\alpha_n$. Substituting (9) and (10) into the previous equation, we obtain

$$\pi_{n+1}^\sigma = \left(1 + \frac{\beta_{n-1}}{\alpha_{n-1}}\alpha_n + \alpha_n \sigma\right)\pi_n^\sigma - \frac{\beta_{n-1}}{\alpha_{n-1}}\alpha_n \pi_{n-1}^\sigma = R_{n+1}(-\sigma). \tag{11}$$

The above formulation is based on the assumption that the seed and shifted systems are $A\boldsymbol{x} = \boldsymbol{b}$ and $(A + \sigma I)\boldsymbol{x} = \boldsymbol{b}$. Similarly, it can be readily generalized to solve m shifted linear systems $(A + \sigma_i I)\boldsymbol{x}^{(i)} = \boldsymbol{b}$ using the seed system $(A + \sigma_s I)\boldsymbol{x} = \boldsymbol{b}$. The resulting algorithm is given below.

Algorithm 2: Shifted COCG

$\boldsymbol{x}_0^{(i)} = \boldsymbol{p}_{-1}^{(i)} = \boldsymbol{0},\ \boldsymbol{r}_0 = \boldsymbol{b},$

$\beta_{-1} = 0,\ \pi_0^{(s,i)} = \pi_{-1}^{(s,i)} = \alpha_{-1} = 1,$

for $n = 0, 1, \ldots$ until $\dfrac{\|\boldsymbol{r}_n\|}{\|\boldsymbol{b}\|} \leq \epsilon_1$ **do:**

  $\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_{n-1}\boldsymbol{p}_{n-1},$

  $\alpha_n = \dfrac{\boldsymbol{r}_n^T \boldsymbol{r}_n}{\boldsymbol{p}_n^T (A + \sigma_s I)\boldsymbol{p}_n},$

  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n \boldsymbol{p}_n,$

  {Begin shifted system}

  for $i(\neq s) = 1, \ldots, m$ **do:**

    if $\|\boldsymbol{r}_n^{(i)}\| > \epsilon_2\|\boldsymbol{b}\|$ **then**

      $\pi_{n+1}^{(s,i)} = R_{n+1}^{(s)}(\sigma_s - \sigma_i), \leftarrow$ (11)

      $\beta_{n-1}^{(i)} = \left(\pi_{n-1}^{(s,i)}/\pi_n^{(s,i)}\right)^2 \beta_{n-1},$

      $\alpha_n^{(i)} = \dfrac{\pi_n^{(s,i)}}{\pi_{n+1}^{(s,i)}}\alpha_n,$

      $\boldsymbol{p}_n^{(i)} = \dfrac{1}{\pi_n^{(s,i)}}\boldsymbol{r}_n + \beta_{n-1}^{(i)}\boldsymbol{p}_{n-1}^{(i)},$

      $\boldsymbol{x}_{n+1}^{(i)} = \boldsymbol{x}_n^{(i)} + \alpha_n^{(i)}\boldsymbol{p}_n^{(i)},$

    end if

  end

  {End shifted system}

  $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n - \alpha_n(A + \sigma_s I)\boldsymbol{p}_n,$

  $\beta_n = \dfrac{\boldsymbol{r}_{n+1}^T \boldsymbol{r}_{n+1}}{\boldsymbol{r}_n^T \boldsymbol{r}_n}.$

end

Shifted COCG with seed switching technique

We can see from Algorithm 2 that if $|\pi_n^{(s,i)}| = |R_n^{(s)}(\sigma_s - \sigma_i)| \geq 1$, then $\|\boldsymbol{r}_n^{(i)}\| \leq \|\boldsymbol{r}_n\|$. Hence, if we could find a seed system such that $|R_n^{(s)}(\sigma_s - \sigma_i)| \geq 1$, then all shifted systems could be solved. However, it is extremely hard to find such system in *a priori* except some special cases discussed in [2]. In this section, we will avoid such problem by using the following strategy:

  I. Choose a seed system, and then start Algorithm 2;

  II. If the seed system was solved at nth iteration, then find the new one;

  III. Start Algorithm 2 from $(n + 1)$th iteration using the new seed system.

In II, as one of criteria for choosing the new seed system $\tilde{s}$, we adopt $\tilde{s} = \arg\max_{i \in I}\{\|\boldsymbol{r}_n^{(i)}\|\}$, where I denotes an index set of unsolved systems. In III, we need two steps to switch the old seed system to the new one. First, compute

$$\pi_{n+1}^{(s,\tilde{s})} = R_{n+1}^{(s)}(\sigma_s - \sigma_{\tilde{s}}), \quad \beta_n^{(\tilde{s})} = \left(\pi_n^{(s,\tilde{s})}/\pi_{n+1}^{(s,\tilde{s})}\right)^2 \beta_n$$

for obtaining $\boldsymbol{r}_{n+1}^{(\tilde{s})}$ and $\beta_n^{(\tilde{s})}\boldsymbol{p}_n^{(\tilde{s})}$. Since it follows from $\boldsymbol{r}_{n+1}^{(\tilde{s})} + \beta_n^{(\tilde{s})}\boldsymbol{p}_n^{(\tilde{s})}$ that we have $\boldsymbol{p}_{n+1}^{(\tilde{s})}$, we can start COCG solving the system $(A + \sigma_{\tilde{s}}I)\boldsymbol{x}^{(\tilde{s})} = \boldsymbol{b}$ from
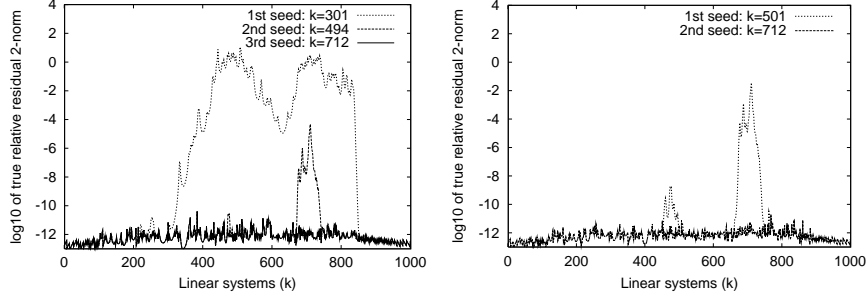
4

Figure 1: The true relative residual 2-norm histories after each seed switching finished. The first seed is $k = 301$ on the left and $k = 501$ on the right.

$(n+1)$th iteration step. Second, to solve remaining systems by using the new seed $\tilde{s}$, it requires generating $\alpha_{n+1}^{(i)}$, $\beta_n^{(i)}$ from the new seed. We see that they can be readily generated by the following polynomial:

$$\pi_{n+1}^{(\tilde{s},i)} = R_{n+1}^{(\tilde{s})}(\sigma_{\tilde{s}} - \sigma_i) \quad \text{for all} \ \ i \in I.$$

To obtain the above polynomial, we need to compute

$$\alpha_i^{(\tilde{s})} = \left(\pi_i^{(s,\tilde{s})} / \pi_{i+1}^{(s,\tilde{s})} \right) \alpha_i, \quad \beta_j^{(\tilde{s})} = \left(\pi_j^{(s,\tilde{s})} / \pi_{j+1}^{(s,\tilde{s})} \right)^2 \beta_j$$

for $i = 0, \ldots, n$, $j = 0, \ldots, n-1$. Hence, the switching strategy requires only scalar operations, and moreover we can see that if breakdown does not occur, iterating the process from (II) to (III) enables us to keep solving the systems without losing the dimension of the Krylov subspace that has been generated until the last switching.

4 Numerical examples

In this section, we report the results of numerical experiments. The problem originally comes from [3] and is written as follows:

$$(\sigma_k I - H)\boldsymbol{x}^{(k)} = \boldsymbol{e}_1, \quad k = 1, \ldots, m,$$

where, $\sigma_k = 0.4 + (k - 1 + i)/1000$, $H \in R^{2048 \times 2048}$ is a symmetric matrix, $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^T$, and $m = 1001$. Since $(\sigma_k I - H)$ is complex symmetric, the iterative solvers we used are COCG and shifted COCG. We can also apply shifted Bi-CGSTAB(ℓ)[2] and GMRES [1] to the above problem since they can be used for general non-Hermitian shifted linear systems. However, they do not exploit the property of complex symmetric matrix. This leads to high computational costs per iteration step.

All experiments were performed on an ALPHA work station with a 750MHz processor using double precision arithmetic. Code were written in Fortran 77

5

Table 1: Numerical results of COCG and shifted COCG.

Initial seed	Switching	Total MVs
$k = 301$	2	330
$k = 501$	1	328
COCG	–	124396

and compiled with the optimization option -$O4$. The switching criterion is $\tilde{s} = \arg\max_{i \in I}\{\|\boldsymbol{r}_n^{(i)}\|\}$. The stopping criteria are $\epsilon_1, \epsilon_2 \leq 10^{-12}$. We report two examples for $k = 301$ and $k = 501$ as a initial seed system. True residual 2-norm histories are given in Fig. 1.

In Fig. 1 on the left, 518 systems remained unsolved when the first seed system $k = 301$ converged. Then, the next seed $k = 494$ solved 412 more systems. Finally, the third seed $k = 712$ solved all of the remaining systems. In Fig. 1 on the right, we chose $k = 501$ as an initial seed system. This choice led to 147 unsolved systems. Finally, the next seed $k = 712$ solved all of the remaining systems.

Numerical results are shown in Table 1. Total MVs denotes the total number of matrix-vector multiplications. We can see from Table 1 that shifted COCG required only about 0.27% of Total MVs of COCG.

5   Concluding remarks

Since the original problem (1) is a fundamental quantum-mechanical equation, the present method is applicable, in principle, to various nanoscale materials, such as silicon, carbon, metals, polymers and so on, if the Hamiltonian for electrons is given as an explicit matrix H.

The present paper gives an interdisciplinary research between mathematics and physics,which shows that the computational science can give an important contribution to nanoscience through the development of general numerical algorithms, when a fundamental equation is formulated in physics.

References

[1] Datta B, Saad Y (1991) Lin Alg Appl 154-156:225-244

[2] Frommer A (2003) Computing 70:87-109

[3] Takayama R, Hoshi T, Sogabe T, Zhang S.-L., Fujiwara T (cond-mat/0503394)

[4] van der Vorst HA, Melissen JBM (1990) IEEE Trans Mag 26(2):706-708